

*AB*

primitive. Once all of the dependence bits in a candidate's dependence vector have been cleared, that candidate is then said to be eligible and can then pass to a graphics accelerator/rasterizer on a subsequent clock cycle.

---

*In the Claims:*

*|||||*

Please cancel claims 2-3, 5-6, and 10 and amend claims 1, 8-9, 11-12, 14-16, and 22 as follows.

- SUBJ*
- AM*
- SUBJ AS*
- AM*
- SUBJ*
- AM*
1. (Once Amended) A method for determining dependencies between a first graphics primitive and a second graphics primitive, the method comprising:
    - calculating a first bounding box for the first graphics primitive, the first bounding box surrounding at least one source operand of the first graphics primitive;
    - calculating a second bounding box for the second graphic primitive, the second bounding box surrounding a destination operand of the second graphic primitive; and
    - determining whether the first bounding box and the second bounding box overlap, wherein a dependency is detected if the bounding boxes overlap.
  4. (Once Amended) The method of claim 1, wherein a write after read dependency is detected if the second bounding box overlaps the first bounding box.
  7. (Once Amended) The method of claim 1, wherein a read after write dependency is detected if the first bounding box overlaps the second bounding box.
  8. (Once Amended) A method for determining whether a dependency exists between a first graphics primitive and a second graphics primitive, comprising:
    - comparing a set of destination pixel locations of the first graphics primitive with a set of destination pixel locations of the second graphics primitive; and
    - determining whether a dependency exists between the first and second graphics primitives as a function of the comparison of the destination pixel locations of the first and second graphic primitive.

*AC*

9. (Once Amended) The method of claim 8 wherein the step of comparing further comprises:

calculating a first bounding box which surrounds the set of destination pixel locations of first graphics primitive;

calculating a second bounding box which surrounds the set of destination pixel locations of the second graphics primitive; and

comparing the first bounding box with the second bounding box.

---

*SUBS1*

11. (Once Amended) A method for determining whether a dependency exists between a first graphics primitive and a second graphics primitive, comprising:

comparing a set of destination pixel locations of the first graphics primitive with at least one set of source pixel locations of the second graphics primitive; and

determining whether a dependency exists between the first and second graphics primitives as a function of the comparison.

*AM*

12. (Once Amended) The method of claim 11 wherein the step of comparing further comprises:

calculating a first bounding box which surrounds the set of destination pixel locations of the first graphics primitive;

calculating a second bounding box for each of the at least one set of source pixel locations of the second graphics primitive; and

determining whether there is dependency if the first and second bounding boxes overlap.

---

*SUBS2*

14. (Once Amended) An apparatus for detecting dependencies between a first graphics primitive and a second graphics primitive, comprising:

a destination reservation station for storing a destination bounding box location for the first graphics primitive;

a source reservation station for storing a source bounding box location for the first graphics primitive; and

*AG*

a first comparator for comparing the destination bounding box location for the first graphics primitive with a bounding box location of the second graphics primitive and generating a first resultant bit.

15. (Once Amended) The apparatus of claim 14 further comprising:  
a second comparator for comparing the source bounding box location for the first graphics primitive with the destination bounding box location of the second graphics primitive and generating a second resultant bit.

16. (Once Amended) The apparatus of claim 15 further comprising:  
a third comparator for comparing the destination bounding box location for the first graphics primitive with a source bounding box location of the second graphics primitive and generating a third resultant bit.

- Subj 1*  
*Aey*
22. (Once Amended) The method of claim 20 wherein the step of detecting is comprised of:  
comparing the destination region bounding box coordinates for the primitives in the at least two which have not yet been completely processed with the destination region bounding box coordinates and the source region bounding box coordinates for the next primitive to be processed in order to detect a dependency; and  
detecting a dependency if there is an overlap in the destination region bounding box coordinates for the primitives in the at least two which have not yet been completely processed and either the destination region bounding box coordinates or the source region bounding box coordinates for the next primitive to be processed.

*Subj:*

*AP*

---

Please add the following claims:

28. (New) A method for determining dependencies between a first graphics primitive and a second graphics primitive, the method comprising:
- calculating a first bounding box for the first graphics primitive, the first bounding box surrounding a destination operand of the first graphics primitive;
- calculating a second bounding box for the second graphic primitive, the second bounding box surrounding a destination operand of the second graphic primitive; and
- determining whether the first bounding box and the second bounding box overlap, wherein a dependency is detected if the bounding boxes overlap.
- 
29. (New) The method of claim 28, wherein a write after write dependency is detected if the bounding boxes overlap.